



## Mejorando el mantenimiento de código

### Transcripción

[00:00] Bueno amigos, vamos a continuar con nuestro código. Habíamos aprendido a crear variables dentro de JavaScript. En nuestro código el resultado era esto de aquí. Habíamos calculado la edad que tenía Juan, Pedro, Carlos en función de su fecha de nacimiento, y Jimena, Paola, Laura, y calculamos el promedio de las edades.

[00:27] Vamos a continuar trabajando con esto de variables y cómo hacer que nuestro código sea más fácil de realizar mantenimiento. Para repasar esos conceptos, vamos a eliminar aquí una parte del código para que quede un poco más corto toda esa parte, todos esos saltos de línea. Voy a salvar y vamos a ver cómo quedó.

[01:01] Tenemos que Juan, Pedro y Carlos. Aquí solo recordando, fíjense que para los saltos de línea, teníamos que colocar todo ese código para cada salto de línea, dentro de JavaScript, si quiero hacer un salto de línea tengo que pasarle la función document write, y en este caso el BR, abriendo y cerrando menor y mayor, consigno mayor y menor, significa un salto de línea.

[01:33] Ahorita tenemos dos saltos de línea. Imagínense que en el día nuestro jefe o la persona que nos está solicitando nuestro programa nos diga: "Christian, yo no quiero dos saltos de línea, quiero tres en cada uno de los documentos, cada una de las líneas de código". Ahí yo voy a tener que agregar eso, salvo con "Ctrl + S", voy allá, y ahora tengo tres saltos de línea.

[02:06] El día de mañana nuestro jefe nos vuelve a decir: "Ahora quiero cuatro".

Voy a tener que agregar un document write BR en cada uno de los lugares, guardar y actualizar mi página. Eso vuelve el trabajo un poco tedioso. Por un lado da la posibilidad de que tengamos un error y vuelve un trabajo totalmente ineficiente desde el punto de vista de la programación y de darle mantenimiento y realizar modificaciones que naturalmente van a venir.

[02:50] Entonces una de las formas de solucionar es usando variables que ya la habíamos visto. Yo puedo crear aquí una variable que se llame salto de línea, y justamente le voy a pasar la cantidad de salto de líneas que quiero hacer. En este caso le voy a pasar un salto de línea. Entonces, le estoy diciendo que salto de línea es BR. Y aquí ya no voy a escribir BR. Voy a escribir salto de línea.

[03:30] Voy a eliminar esto, voy a eliminar esta parte, y salto de línea. Entonces mi objetivo es que lo que el código, cuando el compilador lea esta variable va a ver que es un salto de línea y va a ejecutar un salto de línea. Voy a guardar, vuelvo a mi programa, actualizo y tengo un salto de línea. El día de mañana mi jefe me dice: "Christian, quiero dos saltos de línea", entonces ya no necesito ir y cambiar en varios lugares.

[04:10] Cambio en un lugar, solo en un lugar, coloco dos saltos de línea, guardo aquí, actualizo, dos saltos de línea. El día de mañana la persona que me solicitó el código me dice: "Christian quiero tres saltos de línea". Solo cambio en un lugar, guardo, vuelvo al programa, tres saltos de línea.

[04:41] ¿Entonces qué es lo que me permiten? ¿Qué es lo que me ayuda a usar las variables? A no tener que repetir el código un millón de veces, y cada vez que yo tenga una modificación lo hago en un solo lugar. Con este concepto de usar variables, reforzando un poco el uso de variables y la facilidad que me permite en el futuro de realizar mantenimiento.

[05:11] Volviendo solo a reforzar un poco el tema de la convención de la nomenclatura en nuestras variables. Aquí yo no coloqué X. ¿Podría haberlo hecho? Sí. ¿Va a funcionar? Va a funcionar, pero el día de mañana yo veo este código y ¿qué significa X? Voy a tener que interpretar.

[05:28] Y peor aún, si una tercera persona, si otra persona ve mi código, y si no es una persona que conoce o no tiene conceptos de programación, va a ser hasta más difícil entender lo que ese código está haciendo. Entonces, como convención de programación y como buenas prácticas, vamos a utilizar siempre nombres de variables que sean representativos, que sean explícitos de lo que están haciendo, en este caso es un salto de línea.

[05:57] Vamos a continuar en la otra sección, explorando un poco más el tema de variables.